

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Appellant/Applicant : Sabin Belu Confirmation No. : 8657
Application No. : 09/818,134 Group Art Unit : 2166
Filing Date : March 27, 2001 Examiner : Channavajjala,
Srirama T.
Docket No. : RN058 (2635-019-03) Customer No. : 72455
Title : **SYSTEMS AND METHODS FOR CREATING SELF-
EXTRACTING FILES**

**REPLY BRIEF
37 C.F.R. § 41.41**

TO THE COMMISSIONER FOR PATENTS

This is a reply brief pursuant to 37 C.F.R. § 41.41 from the decision of the Examiner, dated July 6, 2009, finally rejecting claims 1-34, and the Examiner's Answer as mailed on April 19, 2010 of the above-referenced patent application.

This Reply Brief is filed within two (2) months of the Examiner's Answer as mailed April 19, 2010 pursuant to 37 C.F.R. § 41.39.

**37 CFR §1.8
CERTIFICATE OF TRANSMISSION**

I hereby certify that this correspondence is being transmitted via the Office electronic filing system, EFS-Web, addressed to the Commissioner for Patents, PO Box 1450, Alexandria, Virginia 22313-1450 on the date indicated below.

/Lisa Napoli/
Lisa Napoli

June 21, 2010
Date

(I) Argument

In the Examiner's Answer dated 4/19/2010, the Examiner mostly repeated prior rejections. For the convenience of The Board, the Applicant's agent carries forward arguments previously presented below headings "**Previous Arguments**", below. In addition to repeating old reasons for rejection, the Examiner also introduced new locations within the cited documents. Responses to the newly cited locations are inserted under sections headed "**Reply Brief Arguments**", below.

In the Notice of Panel Decision mailed 1/12/2010, only claims 1-10, 20, 24, 26, 27, and 31-33 are listed as rejected. However, claims 1-34 are pending in the application, and claims 1-34 were listed as rejected in the Advisory Action mailed 9/14/2009 and earlier communications. A telephone call was placed and voicemail left for the Examiner on 2/8/2010 to request clarification, but no response was received.

The Applicants' agent herein provides reasons for allowability for all of pending claims 1-34.

Claims 1-10, 20, 24, 26, 27, and 31-33 are rejected under 35 U.S.C § 102(e) as being anticipated by Halpern et al. [hereafter Halpern], U.S. Patent No. 6,282,711, filed on Aug. 10, 1999.

Claim 1

Claim 1 recites "receiving from the user enabled electronic device, an input file to be used in creating a self-extracting file; and without further action by the user enabled electronic device, creating a self-extracting file using the input file, wherein the input file is configured to be automatically launched upon execution of the self-extracting file."

Reply Brief Arguments

The Examiner, in his Answer to the Appeal Brief, cites Halpern column 4, lines 9-18. [Examiner's Reply, p. 17, last paragraph – p. 18, second paragraph] This section of Halpern discloses:

"For a self-extracting executable containing decompression and auto-start utilities, the user simply executes the received setup.exe or install.exe file. This executable may also include a client installer program, which may be merely a cloned copy of the

installer set generator program. The client installer program may be configured to permit further user interaction, or if selections have already been made at the server level, may just install the contents of the package without further user intervention.”

Contrary to the Examiner’s contention, this does not “exactly read on applicant’s definition of ‘self-extracting’ is a ‘executable file’.” Moreover, this section of Halpern does not disclose the recitation of claim 1.

As indicated, claim 1 recites, "receiving from the user enabled electronic device, an input file to be used in creating a self-extracting file; and without further action by the user enabled electronic device, creating a self-extracting file using the input file, wherein the input file is configured to be automatically launched upon execution of the self-extracting file."

For reasons similar to those given previously, Halpern does not disclose “receiving from the user enabled electronic device, an input file,” as recited by claim 1. The self-extracting file disclosed by Halpern is delivered from a server to a client, not the other way around.

Halpern also does not disclose, at this location, “without further action by the user enabled electronic device, creating a self-extracting file using the input file.” To the contrary, Halpern discloses receiving a self-extracting file. [emphasis added] Moreover, Halpern does not disclose even receiving a self extracting file containing an input file received from the user enabled electronic device. Rather, the self extracting file contains an input file received from a server.

Halpern also does not disclose, at this location, “the input file is configured to be automatically launched upon execution of the self-extracting file,” as recited by claim 1. “The input file” is not an installer. It cannot be an installer because the Examiner elsewhere (incorrectly) equates selecting program options to providing an input file. Halpern’s program options are not disclosed anywhere to pertain to an installer, but rather to an application program that the installer installs. Therefore, to disclose the recitation of claim 1, Halpern would have to disclose both launching an installer program and launching the program that was installed by the installer program. Halpern does not disclose this. Halpern merely discloses self-extracting and installing, not self-extracting, installing, and launching the installed program.

Accordingly, the Examiner still has not shown Halpern to disclose all the limitations of claim 1.

Responses to all others of the Examiner's reasons for rejection are included below in the Previous Arguments.

Previous Arguments

Halpern does not disclose receiving an input file to be used in creating a self-extracting file from a user-enabled electronic device.

Contrary to the Examiner's contention, Halpern [C:5; L: 41-44], does not disclose receiving an input file from a user-enabled electronic device. Rather, Halpern discloses providing one of three selected user interfaces, UI-1, UI-2, or UI-3 to the user for selecting application components and options, followed by subsequent delivery of a file to the user by a server.

Even if one (incorrectly) equates selection an input file through Halpern's components and options interface to receiving an input file from a user-enabled electronic device (which is not shown in Halpern), Halpern still would not disclose this limitation because Halpern's components and options do not have a 1:1 correspondence with files.

Following Halpern's selection of components and options, an "options manager" retrieves metadata from a database and an "installer set generator" accesses a component pool to produce a custom set of files corresponding to the user's selections [C: 7; L: 23-38]. Thus, the selected components and options are program functionalities that are used as input to the options manager, and do not represent files.

Accordingly, Halpern does not disclose receiving an input file from the user enabled electronic device, as recited by claim 1.

Halpern does not disclose creating a self-extracting file using the input file without further action by the user enabled electronic device.

Even if user selection of "components and options" is incorrectly assumed equivalent to receiving an input file, Halpern is, at best, silent regarding further action. Halpern does not disclose whether or not further action is required by the user, or by the user enabled electronic device.

Halpern does not disclose the input file is configured to be automatically

launched upon execution of the self-extracting file.

Even if one incorrectly assumes Halpern discloses receiving an input file from a user-enabled electronic device, and assumes that no further action is required by the user-enabled electronic device, Halpern still does not disclose that the input file is configured to be automatically launched upon execution of the self-extracting file.

Halpern discloses that the compressed files are configured to be automatically extracted. [emphasis added] Automatic extraction of a file is not the same as automatically launching the file after it is extracted.

[Previous argument removed in Reply Brief.]

With respect to the Examiner's contention that Halpern discloses "auto-start", [C: 3; L: 42-49; and Fig. 1] there is no disclosure of anything that bears any relation to the recited limitations of claim 1. Halpern does not mention anything that could remotely be interpreted as "auto-start" at the cited location. The Applicant's agent searched the entirety of Halpern and found that there is a mention of an "auto-start utility" [C: 6; L: 49], but this is not related to the recitation of claim 1. Halpern apparently only includes this "auto-start utility" in a list of possible program components and does not describe what it is. It is suggested that perhaps this relates to a program component that can automatically launch an application responsive to clicking on a linked data file type. But this does not mean that the compressed file is automatically launched upon extraction. Nowhere does Halpern disclose an input file that is "configured automatically launched upon execution of the self-extracting file," as recited by claim 1 and, as explained above, implies the opposite.

Generally, the Examiner seems to not give proper weight to the claim language "wherein the input file is configured to be automatically launched upon execution of the self-extracting file." A common thread through the citations provided by the Examiner is that extraction will occur automatically upon execution of a self-extracting compressed file. In contrast, none of the cited art discloses or reasonably suggests that an input file is not only extracted, but also launched for execution upon extraction.

Accordingly, Halpern does not disclose all the limitations of claim 1, and claim 1 is allowable over Halpern.

Claim 2

Claim 2 recites automatically generating a filename for the self-extracting based in part on the filename of the received input file.

Claim 2 is allowable by virtue of its dependency from claim 1.

Moreover, Halpern does not disclose naming a self-extracting file based in part on the filename of a received input file (even if, *arguendo*, the input file was received from the user enabled electronic device). Halpern is apparently silent on the names of input files, and also does not disclose naming a self-extracting file based in part on the name of an input file.

Claim 2 is also allowable over Halpern for at least this additional reason.

Reply Brief Arguments

The Examiner, in his Answer, indicates something about file types used by Halpern and names “setup.exe”, “install.exe”. [Examiner’s Reply, p. 19, paragraph e) and following paragraph] Respectfully, the Applicant’s agent cannot understand the Examiner’s argument.

If the Examiner is trying to say that the .exe extension is used in an input file and the compressed file, then the Applicant’s agent respectfully points out that a file extension is not a filename. The filename is located to the left of the dot.

If the Examiner is trying to say that “install.exe” or “setup.exe” are input files (and also used as names for the self-extracting file), then the Applicant’s agent strongly disagrees. Even if the compressed contents were named “install” or “setup”, Halpern does not disclose this. Moreover, it is respectfully submitted that “install” or “setup” are not “an input file” received “from the user enabled electronic device,” as recited by claim 1; and thus Halpern does not disclose “wherein the received input file has an associated filename and wherein a filename for the self-extracting file is configured to be automatically generated based in part on the associated filename of the received input file,” as recited by claim 2.

Previous Arguments

Claim 3

Claim 3 recites “receiving an input file to be used in creating a self-extracting file, wherein the file is one of a plurality of file types; and in response to only a single action, creating a self-extracting file from the input file, wherein the input file is configured to be automatically launched upon execution of the self-extracting file.”

Claim 3 is allowable for reasons similar to those given for claim 1. Namely, Halpern does not disclose the input file is configured to be automatically launched upon execution of the self-extracting file.

Halpern discloses that the compressed files are configured to be automatically extracted. [emphasis added] Automatic extraction of a file is not the same as automatically launching the file after it is extracted.

Moreover, by comparing Halpern’s Step 7 to Step 12 [C: 7; L: 55-56], it is apparent that Halpern’s “*input file*” is not even automatically *installed* upon execution of the self-extracting file. The client installer program is a separate entity, as evidenced by Halpern’s indication that “the user executes the retrieved file and runs the client installer.” [emphasis added] A program that is not even installed automatically after extraction also cannot be launched automatically after extraction, as recited by claim 1. This view is further strengthened by reference to Halpern [C: 6; L: 47-52], which contains its own rebuttal of the Examiner’s position: “the user may simply execute the received setup.exe or install.exe file to immediately install the applications and options...” Requiring a user to execute a program (setup.exe or install.exe) is inconsistent with automatic launching upon execution of the self-extracting file (even if it is an input file received from the user enabled electronic device that is launched).

With respect to the Examiner’s contention that Halpern discloses “auto-start”, [C: 3; L: 42-49; and Fig. 1] there is no disclosure of anything that bears any relation to the recited limitations of claim 1. Halpern does not mention anything that could remotely be interpreted as “auto-start” at the cited location. The Applicant’s agent searched the entirety of Halpern and found that there is a mention of an “auto-start utility” [C: 6; L: 49], but this is not related to the recitation of claim 1. Halpern apparently

only includes this “auto-start utility” in a list of possible program components and does not describe what it is.

Accordingly, Halpern does not disclose all the limitations of claim 3, and claim 3 is allowable over Halpern.

Claims 4-7

Claims 4 - 7 each recite a type of single action that initiates creation of a self-extracting file.

Since Halpern does not disclose creating a self-extracting file from only a single action, it also does not disclose the types of single actions by a user recited by claims 4 - 7. Claims 4 - 7 are allowable by virtue of their dependence from claim 3 and are also allowable for this additional reason.

Claim 8

Claim 8 is allowable by virtue of its dependence from claim 3.

Claim 9

Claim 9 recites “generating a filename for the self-extracting file, wherein the generated filename is based on a filename associated with the input file.”

Halpern is silent with respect to how the name of the self-extracting file may be based or not be based on the name of an input file.

Claim 9 is allowable by virtue of its dependence from claim 3 and is additionally allowable for reasons similar to those given for claim 2.

Claim 10

Claim 10 is allowable for reasons similar to those given for claim 1. Namely, Halpern does not disclose “automatically creating a self-extracting file configured to automatically launch the received input file responsive to execution of the self-extracting file,” as recited by claim 10.

Claim 20

Claim 20 recites “in response to only a single action, creating a self-extracting file from an input file, wherein the input file is one of a plurality of file types, and automatically selecting a loader based on the input file's type, and wherein the input file will be automatically launched upon execution of the self-extracting file.”

As discussed above with respect to claim 1, Halpern does not disclose an input file being automatically launched upon execution of a self-extracting file.

Moreover, Halpern does not disclose automatically selecting a loader based on the input file's type. The Examiner has asserted that “the limitations of claims 10, 20, 26 and 32 are rejected in the analysis of Claim 1 above, and these claims are rejected on that basis.” However, claim 1 does not recite automatically selecting a loader based on an input file's type.

Claim 20 is additionally allowable for reasons similar to those given for claim 1 with respect to “wherein the input file will be automatically launched upon execution of the self-extracting file.”

Halpern does not disclose “automatically selecting a loader based on the input file's type,” and moreover does not disclose “wherein the input file will be automatically launched upon execution of the self-extracting file,” both of which are recited by claim 20.

Accordingly, Halpern does not disclose all the limitations of claim 20, and claim 20 is allowable over Halpern.

Claim 24

Claim 24 recites a “a third module for creating, in response to only a single action by a user, an executable file from the compressed input file, wherein the input file will be automatically launched upon execution of the executable file.”

For reasons similar to those given for claim 1, Halpern does not disclose a “module for creating, in response to only a single action by a user, an executable file from the compressed input file, wherein the input file will be automatically launched upon execution of the executable file.”

Halpern only discloses self-extraction, and does not disclose automatic launching of an input file following the self-extraction. Accordingly, Halpern does not disclose all the limitations of claim 24, and claim 24 is allowable over Halpern.

Claim 26

Claim 26 recites “receiving an input file from a user to be used in creating a self-extracting file, wherein the input file is of any file type, and automatically creating a self-extracting file.”

In contrast, and as discussed above with respect to claim 1, Halpern does not disclose receiving from a user an input file to be used in creating a self-extracting file. Halpern discloses that a user is allowed to select a subset of program “components and options” available for installation, after which a subset of files corresponding to those components and options are packaged and distributed to the user. Halpern does not disclose receiving any file from a user. Accordingly, Halpern does not disclose all the limitations of claim 26, and claim 26 is allowable over Halpern.

Claim 27

Claim 27 recites, “receiving, in response to a single action, an input file to be used in creating an executable file, wherein the input file is one of a plurality of file types; and without further instructions, creating an executable file using the received input file, wherein the executable file includes a compressed copy of the input file, and wherein the compressed copy of the input file is automatically decompressed and launched upon execution of the executable file.”

For reasons described above with respect to claim 1, Halpern does not disclose “wherein the compressed copy of the input file being automatically decompressed and launched upon execution of an executable file.”

Moreover, for reasons similar to those given for claim 1, Halpern also does not disclose receiving an input file in response to a single action.

Accordingly, Halpern does not disclose all the limitations of claim 27, and claim 27 is allowable over Halpern.

Claim 31

Claim 31 recites, “in response to a first action, creating an executable file from an input file, wherein the executable file includes a compressed copy of the input file, and wherein the executable file includes code to decompress and to load the compressed input file; and in response to a second action, executing the executable file to decompress the compressed copy of the input file and launching the decompressed input file with appropriate application software.”

In contrast, as discussed above with respect to claim 1, Halpern does not disclose “creating an executable file from an input file, wherein the executable file includes a compressed copy of the input file, and wherein the executable file includes code to decompress and to load the compressed input file.” This specific disclosure is nowhere to be found in Halpern. Moreover, for reasons given for claim 1, Halpern does not generally disclose an “input file is configured to be automatically launched upon execution of the self-extracting file,” and therefore also does not disclose the executable file includes “code to decompress and to load the compressed input file,” [emphasis added] as recited by claim 31.

Accordingly, Halpern does not disclose all the limitations of claim 31, and claim 31 is allowable over Halpern.

Claim 32

Claim 32 recites “receiving, in response to a single action, an input file to be used in creating a self-extracting file; without further instruction, creating a self-extracting file using the input file and automatically launching the input file upon execution of the self-extracting file.”

For reasons described above with respect to claim 1, Halpern does not disclose “receiving, in response to a single action, an input file to be used in creating a self-

extracting file; without further instruction, creating a self-extracting file using the input file.” Halpern is silent with respect to whether further instruction is required to create a self-extracting file.

Moreover, Halpern does not disclose “automatically launching the input file upon execution of the self-extracting file.” Halpern only discloses self-extraction.

Accordingly, Halpern does not disclose all the limitations of claim 32, and claim 32 is allowable over Halpern.

Claim 33

Claim 33 is allowable by virtue of its dependence from claim 32, and for at least the reasons given for claim 32.

Claims 21-23 are rejected under 35 U.S.C § 102(e) as being anticipated by Wygodny et al. [hereafter Wygodny], U.S. Patent No. 6,202,199 based on provisional application No. 60/055,165 filed on July 31, 1997.

Claim 21

Claim 21 recites, “displaying a first frame used to allow a user to specify an input file to be converted to a self-extracting file; receiving the input file specified by the user, wherein the received input file is automatically configured as a self-extracting file, and wherein the input file is automatically launched upon execution of the self-extracting file; and displaying a second frame, wherein the second frame includes a link related to the self-extracting file created from the user specified input file.”

Wygodny does not disclose “the input file is automatically launched upon execution of the self extracting file.”

Wygodny discloses: “At the end of installation, the user 110 can launch the agent 102.” [C: 17; L: 1-2] However, if Wygodny’s agent 102 was automatically launched upon execution of the self-extracting file, then the agent would already be

launched. Therefore Wygodny, far from disclosing all the limitations of claim 21, teaches away from “the input file is automatically launched upon execution of the self extracting file,” as recited by claim 21.

Moreover, Wygodny does not disclose “displaying a first frame used to allow a user to specify an input file to be converted to a self-extracting file,” as recited by claim 21.

Wygodny does not disclose “automatically configure[ing] an input file specified by the user as a self-extracting file,” as recited by claim 21.

Wygodny does not disclose “displaying a second frame wherein the second frame includes a link related to the self-extracting file created from the user specified input file,” as recited by claim 21.

Wygodny discloses a system for tracing the execution paths of a software program without requiring modifications to the executable or source code files of that program. As part of this system, Wygodny teaches providing a small executable "agent" program that enables a remote user to generate a trace file, and further teaches that the preferred method of providing this "agent" program includes packaging that program as a self-extracting file. These are not the limitations of claim 21.

The Examiner cited Wygodny's FIG. 3A and “col 8, line 51-55; col 17, line 1-7; fig 3A; fig 9-10” to show displaying a first frame used to allow a user to specify an input file to be converted to a self-extracting file. This is incorrect. Referring to FIG. 3A; [C:8; L:50-57], and [C:9; L: 9-13]; Wygodny's frame window 300 (and specifically, executable pane 314) depicts a particular executable file that is currently being traced. It does not allow a user to specify an input file to be converted to a self-extracting file.

The Examiner also cites [C: 17; L: 1-12] to show that Wygodny discloses automatically configuring an input file specified by the user as a self-extracting file. This is also incorrect. Referring to [C: 17; L: 1-12], Wygodny discloses supplying a tracing "agent" program to a user as a self-extracting "zip file." To install this "agent" program, the user can simply double-click on that zip file. Finally, the user may run the "agent" program, which allows the user to specify both a "Trace Control Information" (TCI) file and a client executable program which the user desires to have traced.

But these are not the limitations of claim 21. Wygodny does not disclose “the

received input file is automatically configured as a self-extracting file.” Moreover, Wygodny does not disclose “wherein the input file is automatically launched upon execution of the self-extracting file,” as recited by claim 21.

The Examiner further cites FIGS. 3A-3B and FIG. 5 to show that Wygodny discloses displaying a second frame that includes a link related to a self-extracting file created from a user-specified input file. This is incorrect. Referring to FIGS. 3B and 5, although Wygodny does teach displaying a second frame (relative to the first frame depicted in FIG. 3A) as part of its user interface, none of the user interface frames depicted include a link related to a self-extracting file that has been created from a user-specified input file.

Accordingly, Wygodny does not disclose all the limitations of claim 21, and claim 21 is allowable over Wygodny. “

Claim 22

Claim 22 recites “a receiving module configured to receive an input file, wherein the input file received is one of a plurality of file types and wherein the input file includes an associated filename; a naming module configured to create and name an output file, wherein the output filename is generated from the associated filename of the input file ...; a self-extracting module configured to transform the output file into a executable file, wherein the self-extracting module receives the input file and the output file from the naming module; a loader module configured to setup the executable file to launch the input file upon execution of the executable file, wherein the loader module receives the executable file and the input file from the self-extracting module; and a compressing module configured to compress the input file and attach the compressed input file to the executable file, wherein the compressing module receives the input file and the executable file from the loader module.”

Wygodny does not disclose receiving an input file, wherein the input file received is one of a plurality of file types. The Examiner cites “col 9, line 9-13, line 57-62, col 12, line 24-35” to show that Wygodny discloses this limitation. However, to the extent that Wygodny discloses any “receiving module,” it does so only with respect to the ability to

receive executable files in order to trace the operations of those executable files. Referring, e.g., to [C: 9; L: 9-62] and [C: 12; L: 24-35], the executable files selectable as input files have a single file format .

Wygodny does not disclose “a naming module configured to create and name an output file, wherein the output filename is generated from the associated filename of the input file.” Applicant’s agent has searched Wygodny and finds no mention of an output filename based on an input filename.

Wygodny does not disclose “a self-extracting module configured to transform the output file into a executable file.” Although the Examiner does not provide a citation to show this limitation, Applicant’s agent has reviewed Wygodny and can find no instance that discloses transforming an output file into an executable file. Referring to [C: 16; L: 41-44] and [C: 17; L: 1-12], Wygodny discloses distributing the tracing agent program to the user as a self-extracting zip file. That tracing agent program, however, is not “an output file” as recited by claim 22.

Wygodny does not disclose “a loader module configured to setup the executable file to launch the input file upon execution of the executable file.”

Wygodny’s self extracting file produces a file that is not automatically launched. For reasons similar to those given for claim 21, Wygodny does not disclose a loader module configured to setup the executable file to launch the input file upon execution of the executable file. Wygodny discloses a self-extracting file. The self extracting file produces a file that is not automatically launched. According to Wygodny [C: 17; L: 1-2] “At the end of installation, the user 110 can launch the agent 102.” If the agent was automatically launched upon extraction, it would not be necessary for the user to launch the agent.

Accordingly, Wygodny does not disclose all the limitations of claim 22, and claim 22 is allowable over Wygodny.

Claim 23

Claim 23 is allowable at least by virtue of its dependence from claim 22.

Claims 11-19, 25, 28-29, 30, and 34 are rejected under 35 U.S.C. § 103(a) as being unpatentable over Halpern et al. [hereafter Halpern], U.S. Patent No. 6,282,711 filed on Aug. 10, 1999 as applied to claims 10 and 32, above, and further in view of Gage et al. [hereafter Gage], U.S. Patent No. 5,923,846 published on July 13, 1999.

The Examiner has not made a *prima facie* case for obviousness.

Claims 11-19

Claims 11-19 are allowable by virtue of their dependence from claim 10, and for at least the reasons given for claim 10.

Claim 25

Claim 25 recites a tangible medium carrying computer instructions to cause a computer to “provide a compressed input data portion corresponding to an input data file, ... and provide a self-extracting stub portion, wherein the self-extracting stub portion is automatically attached to the compressed input data portion and the archive header portion, and wherein the self-extracting stub portion includes a decompression engine to decompress the compressed input data portion, and a loader operable to launch the decompressed input data portion with appropriate application software for handling the input data file.”

In contrast, for reasons given with respect to claim 1, Halpern neither discloses nor renders obvious “a loader operable to launch a decompressed input data portion with appropriate application software for handling the input data file,” as recited by claim 25.

Gage discloses a system for posting and downloading textual messages and files from an online bulletin board system. Referring, *e.g.*, to FIG. 5 and corresponding text [C: 14; L: 21-67], Gage teaches storing textual messages as compressed RTF (Rich text format) data having a plurality of compressed portions and an uncompressed header portion. However, Gage does not disclose or reasonably suggest using self-

extracting files as part of its online bulletin board system.

Since Gage does not disclose a self-extracting file, Gage also does not disclose a self-extracting file that also includes “a loader operable to launch the decompressed input data portion with appropriate application software,” as recited by claim 25.

In combination, Halpern and Gage fail to disclose or reasonably suggest “a loader operable to launch the decompressed input data portion with appropriate application software,” as recited by claim 25. The limitation is simply nowhere to be found in either reference.

Accordingly, Halpern and Gage, alone and in combination, fail to disclose or reasonably suggest all the limitations of claim 25, and claim 25 is allowable over Halpern and Gage.

Claim 28

Claim 28 recites a process for producing a computer file, comprising: receiving an input file; automatically opening an output file; automatically adding a decompression engine to the output file for decompressing compressed data; automatically adding loader code to the output file for launching the input file with the appropriate application software for handling the input file; ... [and] automatically compressing the input file according to a data compression method...”

“Automatically compressing the input file” and “automatically adding a decompression engine to the output file for decompressing compressed data” makes the output file “a self-extracting file,” as recited by Claim 1. “Automatically adding loader code to the output file for launching the input file with the appropriate application software for handling the input file” makes the “input file configured to be automatically launched upon execution of the self-extracting file,” as recited by claim 1.

For reasons given above for claim 1, Halpern does not disclose or reasonably suggest “creating a self-extracting file using the input file, wherein the input file is configured to be automatically launched upon execution of the self-extracting file.” Because of the relationship between the recitations of claim 1 and claim 28 described immediately above, Halpern therefore also does not disclose or reasonably suggest

“automatically compressing the input file,...automatically adding a decompression engine to the output file for decompressing compressed data, and ... automatically adding loader code to the output file for launching the input file with the appropriate application software for handling the input file,” as recited by claim 28.

Gage, for reasons given for claim 25, does not disclose or reasonably suggest a self-extracting file that also includes “a loader operable to launch the decompressed input data portion with appropriate application software.” Therefore, Gage also does not disclose or reasonably suggest “automatically adding loader code to the output file for launching the input file with the appropriate application software for handling the input file,” as recited by claim 28.

Accordingly, Halpern and Gage, alone and in combination, do not disclose or reasonably suggest all the limitations of claim 28, and claim 28 is allowable over Halpern and Gage.

Claim 29

Claim 29 is allowable by virtue of its dependence from claim 28 and for at least the reasons given for claim 28.

Claim 30

Claim 30 recites “a method for creating an executable file, comprising: in response to a single action, receiving an input file to be used in creating an executable file, wherein the input file is one of a plurality of file types; and without further instruction, creating an executable file using the received input file, wherein the executable file comprises: a compressed input data portion including data compressed according to a data compression method;...and a stub portion, wherein the stub portion is automatically attached to the compressed input data portion and the archive header portion, and wherein the stub portion includes a decompression engine to decompress the compressed input data portion and a loader to launch the decompressed input data portion.”

For reasons similar to those given above at least with respect to claim 25, Halpern and Gage, alone and in combination, fail to disclose or reasonably suggest “a method for creating an executable file [including a stub portion], wherein the stub portion includes a decompression engine to decompress the compressed input data portion and a loader to launch the decompressed input data portion.”

Accordingly, Halpern and Gage, alone and in combination, fail to disclose or reasonably suggest all the limitations of claim 30, and claim 30 is allowable over Halpern and Gage.

Claim 34

Claim 34 is allowable by virtue of its dependence from claim 32 and for at least the reasons given for claim 32.

For the foregoing additional reasons, the Appellant requests the Board to reverse the Examiner's rejection of claims 1-34, and to rule claims 1-34 allowable or remand the application to the Examiner for allowance of claims 1-34.

Dated this 21st day of June, 2010.

Respectfully submitted,

/CAWiklof/

Christopher A. Wiklof
Registration No. 43,990

Customer No. 72455

Graybeal Jackson LLP
400 - 108th Avenue NE, Suite 700
Bellevue, Washington 98004-5562
Telephone: 425.455.5575
Facsimile: 425.455.1046